PROGRAMACIÓN DECLARATIVA: LÓGICA Y RESTRICCIONES

Grado en Ingeniería Informática / Grado en Matemáticas e Informática

Julio 2016 **Examen Final**

Nombre:

Matrícula:

Todas las preguntas deben comenzar a contestarse en su hoja correspondiente. Pueden añadirse al ejercicio tantas hojas como sean necesarias siempre que estén numeradas y lleven el nombre y número de matrícula del alumno.

EJERCICIO 1 (4 puntos – 35 minutos)

Dos árboles binarios son isomorfos si sus nodos son iguales y sus hijos son a su vez isomorfos (en cualquier orden): el izquierdo de uno isomorfo al izquierdo del otro y el derecho al derecho, o bien el izquierdo de uno isomorfo al derecho del otro y el derecho del primero al izquierdo del segundo. Obviamente, dos árboles iguales son isomorfos.

Dada la estructura de árbol *tree(Raiz,Izqo,Dcho)* donde *Raiz* es el nodo e *Izqo* y *Dcho* los hijos izquierdo y derecho, respectivamente, programar un predicado **puro** *isoB*/2 cierto si y solo si sus dos argumentos son árboles binarios isomorfos. El árbol vacio se representa por *non*.

Ampliando la definición a árboles generales, dos árboles son isomorfos si sus nodos son iguales y cada hijo de uno de ellos es isomorfo a un hijo del otro (y viceversa, o sea, cierto para ambos árboles), en cualquier orden. Dos árboles iguales son isomorfos.

Dada la estructura de árbol *tree(Raiz,Hijos)* donde *Raiz* es el nodo e *Hijos* una lista de los árboles hijos, programar un predicado **puro** *isoG*/2 cierto si y solo si sus dos argumentos son árboles generales isomorfos. El árbol vacio se representa por *non*.

```
isoB(non, non).
isoB(t(R, I1, D1), t(R, I2, D2)):-
      isoB(I1, I2),
      isoB(D1, D2).
isoB(t(R, I1, D1), t(R, I2, D2)):-
      isoB(I1, D2),
      isoB(D1, I2).


isoG(non, non).
isoG(t(R, Hs1), t(R, Hs2)):-
      isoG_hijos(Hs1, Hs2).


isoG_hijos([], []).
isoG_hijos([X|Xs], Ys):-
      isoG_alguno(Ys, X, Ys0),
      isoG_hijos(Xs, Ys0).

isoG_alguno([Y|Ys], X, Ys):-
      isoG(X, Y).
isoG_alguno([Y|Ys], X, [Y|Ys0]):-
      isoG_alguno(Ys, X, Ys0).
```

PROGRAMACIÓN DECLARATIVA: LÓGICA Y RESTRICCIONES

Grado en Ingeniería Informática / Grado en Matemáticas e Informática

Julio 2016 **Examen Final**

Nombre:

Matrícula:

Todas las preguntas deben comenzar a contestarse en su hoja correspondiente. Pueden añadirse al ejercicio tantas hojas como sean necesarias siempre que estén numeradas y lleven el nombre y número de matrícula del alumno.

EJERCICIO 2 (3 puntos – 25 minutos)

Los recursos audiovisuales de una cadena de televisión se dividen en películas y series de televisión. Las películas están representadas por un functor pelicula/2 mientras que las series se representan utilizando el functor serie/2. En ambos casos, el primer argumento es el nombre de la película o de la serie y el segundo argumento es el género de la película o de la serie.

Se pide al alumno programar:

a. Un predicado recursivo **contarRecursos/2** (contarRecursos(L,N)), que dada una lista L con los recursos audiovisuales de una cadena se verifique si N es el número de recursos de dicha cadena. *(0,5 puntos)*

b. Un predicado recursivo **listadoGeneros/2** (listadoGeneros(L,Generos)), que dada una lista L con los recursos audiovisuales de una cadena se verifique si Generos es la lista con los géneros de los recursos de dicha cadena. El predicado debe generar solamente una solución. Codificar además una llamada a dicho predicado que obtenga una lista con los géneros pero sin redundancias. *(1,25 puntos)*

c. Un predicado **listadoNombresPeliculas/2** (listadoNombresPeliculas(L,Peliculas)), que dada una lista L con los recursos audiovisuales de una cadena se verifique si Peliculas es la lista de los nombres de las películas de dicha cadena. El código no debe realizar de manera explícita recorridos de listas. *(1,25 puntos)*

```
(a)
contarRecursos([],0).
contarRecursos([X|R],N):-
      contarRecursos(R,NR),
      N is NR+1.

(b)
listadoGeneros([],[]).
listadoGeneros([X|R],[G|LGeneros]):-
      arg(2,X,G),
      listadoGeneros(R,LGeneros),!.

?- setof(X,(listadoGeneros(LRA,G),member(X,G)),L).

(c)
listadoPeliculas(L,Peliculas):-
      findall(NP, ( member(P,L), P=..[pelicula,NP,_G] ), Peliculas).
```

PROGRAMACIÓN DECLARATIVA: LÓGICA Y RESTRICCIONES

Grado en Ingeniería Informática / Grado en Matemáticas e Informática

Julio 2016 **Examen Final**

Nombre:

Matrícula:

Todas las preguntas deben comenzar a contestarse en su hoja correspondiente. Pueden añadirse al ejercicio tantas hojas como sean necesarias siempre que estén numeradas y lleven el nombre y número de matrícula del alumno.

EJERCICIO 3 (3 puntos – 25 minutos)

Se pide al alumno que escriba un **programa lógico con restricciones en dominios finitos** con la sintaxis de **Ciao Prolog** que calcule todas las soluciones existentes para el criptograma que se presenta a continuación:

AB * C = DE + FG = HI

Cada letra debe reemplazarse por un dígito arábigo diferente del 0 al 9 de tal manera que se cumplan las restricciones arriba reseñadas.

```
:- use_package(fd).
criptograma([A, B, C, D, E, F, G, H, I]) :-
      [A, B, C, D, E, F, G, H, I] in 0..9,
      all_different([A, B, C, D, E, F, G, H, I]),
      (10*A + B) * C .=. (10*D + E) + (10*F + G),
      (10*A + B) * C .=. (10*H + I),
      labeling([A, B, C, D, E, F, G, H, I]).
```